HW3

Ugochi Madibata

11/5/19

Problem 1)

Issues:

Non Terminal Q
- "bbc" and "bc" both have as First(b) = b.
So for evaluating Q we don't know whether it is the first option or the second option based on

Fix:

L ::= Ra | Qba
R ::= aba | caba | RQ
Q = bc

Removing "bbc" from Q made it so there won't the possibility of two path leading from a "b" for Q.

Problem 2)

No, the issue is C. Unless Ac evaluates to "dabac", there will be a situation where both options would start w "c". The same thing happens with B unless Cb evalu to "cB" other wise you'd have a situ where both First(B)s are "d".

Rewrite:

A ::= Ba
B ::= dab | cdab
C ::= cB | daba c

Problem 3)

E ::= ↑L
L ::= ↑↓↓ | ↓
X ::= L↑↓

Proof

First(E) = First(↑) = ↑
First(L) = First(↑) = ↑
First(L) = First(↓) = ↓
First(X) = First(L) = ↑

There is no overlap between the firsts of any option so the grammar is LL(1).

Problem 4.

S ::= S ; S
S ::= d := E
S ::= print(L)
E ::= id
E ::= num
E ::= E + E
E ::= (S , E)
L ::= E
L ::= L , E

S ::= S;S | id := E | print(L)
E ::= id | num | E+E | (S, E)
L ::= E | L, E

S = E | S;S | print(L)

↙

S ::= E | print(L) | S;S
E ::= id | num | E+E | (S, E)

↙

S ::= L | print(L) | S;S
E ::= id | num | (S, E) | E+E
L ::= E | L, E

End Result

S = print(L)
S ::= S;S
E ::= id
E ::= num
E ::= E+E
E ::= (S, E)
L ::= E
L ::= L, E